The convolutional neural networks for Amazigh speech recognition system

Meryam Telmem, Youssef Ghanou
Moulay Ismail University, Meknes, Morocco

Article Info

Article history:

Received May 26, 2020
Revised Sep 13, 2020
Accepted Sep 24, 2020

Keywords:

Amazigh language
Convolutional neural network
Deep learning
Mel frequency cepstral coefficient
Spectrogram
Speech recognition

ABSTRACT

In this paper, we present an approach based on convolutional neural networks to build an automatic speech recognition system for the Amazigh language. This system is built with TensorFlow and uses mel frequency cepstral coefficient (MFCC) to extract features. In order to test the effect of the speaker's gender and age on the accuracy of the model, the system was trained and tested on several datasets. The first experiment the dataset consists of 9240 audio files. The second experiment the dataset consists of 9240 audio files distributed between females and males' speakers. The last experiment 3 the dataset consists of 13860 audio files distributed between age 9-15, age 16-30, and age 30+. The result shows that the model trained on a dataset of adult speaker's age +30 categories generates the best accuracy with 93.9%.

Corresponding Author:

Meryam Telmem
Team TIM, High School of Technology
Moulay Ismail University
Meknes, Morocco
Email: meryamtelmem@gmail.com

1. INTRODUCTION

Deep learning is a branch of machine learning. It consists of learning high-level representations of data using deep neural networks. With technological and scientific advances, deep learning has made a place in many areas especially in the field of automatic speech recognition. Automatic speech recognition is a computer technique intended to transcribe a speech signal into text [1]. Since a long time, the hidden markov models [2, 3] it was a perfect solution to the problems of speech recognition. But, in 2012, deep learning [4] has a revolution with the appearance of convolutional neural network (CNN) [5]. It is arguably that the most popular architecture, they have applications in image and video recognition, recommender systems [6], medical image and audio analysis [7], successfully applied in speech recognition. In this work, we built an Amazigh speach recognition system based on CNN and GPU computation using TensorFlow, which is an open source library written in python and C++ with a model and robust architecture that can be run on multiple CPUs and GPUs. This paper is organized as follows: section 2 we present the related work, section 3 we describe the principle and the theory of speech recognition, section 4 we describe the CNN, section 5 we present TensorFlow. Finally, the section 6 illustrates the experimental results followed by conclusion.

2. RELATED WORK

In our previous work [3] we have developed an Amazigh speech recognition system based on hidden Markov model HMMs using an open source CMU Sphinx-4. The corpus consists of 11220 audio files. The best obtained accuracy was 90% when we have trained our model by using 128 Gaussian mixture models, and 5 number of HMMs states. Palo H K, and *et al.* [8] have determined the age of speaker based on emotional speech prosody and clustering them using fuzzy c-means algorithm. This recognition of speech emotion based on suitable features provides age information that helped the society in different ways. They have used many feature extraction techniques. Among the extracted features, the F0, energy or amplitude, and speech rate.

Zhang H., and *et al.* [9] have studied a series of neural networks based acoustic models; time delay neural network (TDNN), CNNs, and the long short-term memory (LSTM), applied them in the Mongolian speech recognition systems, and compared their performance. The result shows that the LSTM is the most accurate model with 8, 12% WER. Satori H., and *et al.* [10] have developed and Amazigh ASR based on the CMU-Sphinx. The system generated 92.89 % of accuracy. The training was performed by using using 16 Gaussian mixture models. Kumar K., and Aggarwal R. [11] have built a Hindi recognition system using HTK based on the hidden Markov models HMMs. The corpus of training consists of 102 words. The system produced 87.01% of accuracy.

3. AUTOMATIC SPEECH RECOGNITION SYSTEM

The problem of speech recognition aims to convert the speech signal to sequence of observations X, in a process called feature extraction. The decoder looks for the sequence of words W* maximizing the following equation:

$$w^* = \underset{w}{argmax} P(W|X) \tag{1}$$

After applying the Bayes theorem, this equation becomes:

$$w^* = \underset{w}{argmax} \frac{P(X|W)P(W)}{P(X)} \tag{2}$$

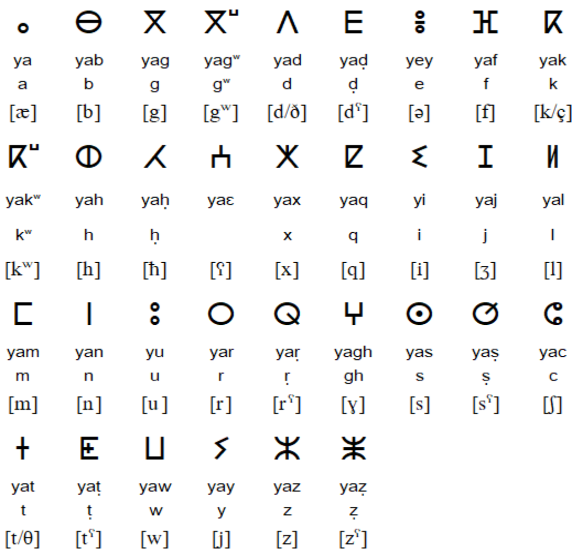
P (X) is considered constant and removed from (2).

$$w^* = \underset{w}{argmax} P(X|W)P(W) \tag{3}$$

3.1. Pre-processing

3.1.1. Audio to spectrum

Speech, whatever its language, is constitute of a finite number of distinctive sound elements. These elements form elementary linguistic units and have the property of changing the meaning of a word. These elementary units are called phonemes [3]. The Phonemes can be seen as the basic elements for coding linguistic information. The Amazigh alphabet contains 33 phonemes [10] as shown in Figure 1.



ⴰ ya a [æ]	ⴱ yab b [b]	ⴳ yag g [g]	ⴳⵯ yagʷ gʷ [gʷ]	ⴷ yad d [d/ð]	ⴹ yaḍ ḍ [dˤ]	ⴻ yey e [ə]	ⴼ yaf f [f]	ⴽ yak k [k/ç]
ⴽⵯ yakʷ kʷ [kʷ]	ⵀ yah h [h]	ⵃ yaḥ ḥ [ħ]	ⵄ yaɛ [ʕ]	ⵅ yax x [x]	ⵇ yaq q [q]	ⵉ yi i [i]	ⵊ yaj j [ʒ]	ⵍ yal l [l]
ⵎ yam m [m]	ⵏ yan n [n]	ⵓ yu u [u]	ⵔ yar r [r]	ⵕ yaṛ ṛ [rˤ]	ⵖ yagh gh [ɣ]	ⵙ yas s [s]	ⵚ yaṣ ṣ [sˤ]	ⵛ yac c [ʃ]
ⵜ yat t [t/θ]	ⵟ yaṭ ṭ [tˤ]	ⵡ yaw w [w]	ⵢ yay y [j]	ⵣ yaz z [z]	ⵥ yaẓ ẓ [zˤ]			

Figure 1. Official table of the tifinaghe alphabet as recommendedby IRCAM [11] has officially been the only writing system for transcribing the Amazight language in Morocco since 2003

The graphic system of the standard Amazighe proposed by the IRCAM comprises [11]

- 27 consonants of: labels (ⴼ, ⴱ, ⵎ), dental (ⵜ, ⴷ, ⵟ, ⴹ, ⵏ, ⵔ, ⵕ, ⵍ).
- The alveolar (ⵙ, ⵣ, ⵚ, ⵥ) (ⵛ, ⵊ) ⴽ, ⴳ, ⴽⵯ, ⴽⵯ, ⵇ, ⵅ, ⵇ, ⵅ, ⵇ, ⵇ ⵇ.
- 2 semi-consonants: ⵢ and ⵡ
- vowels: the full ones (ⴰ, ⵉ, ⵓ), neutral (ⵓ).

CNN takes input an image, so to be able to recognize phonemes it is necessary to pass on spectrum to transform audio into image. This pre-processing phase is the longest and most important phase to build ASR system. In speech recognition system, the most common feature extraction techniques are based on spectrum: PLP, the spectrogram, the mel spectrogram [12], mel frequency cepstral coefficient (MFCC). In this work we have used MFCC technique.

3.1.2. The spectrogram

The spectrogram is a representation of an audio file in a frequency domain. In order to convert raw data to spectrogram we apply short-time fourier transform [13]. The produce matrice is then fed into a multi-layer CNN followed with a fully-connected with softmax activation which generates the classification vector. The following Figure 2 lists the spectrogram of the alphabet ya, yab, and yad:

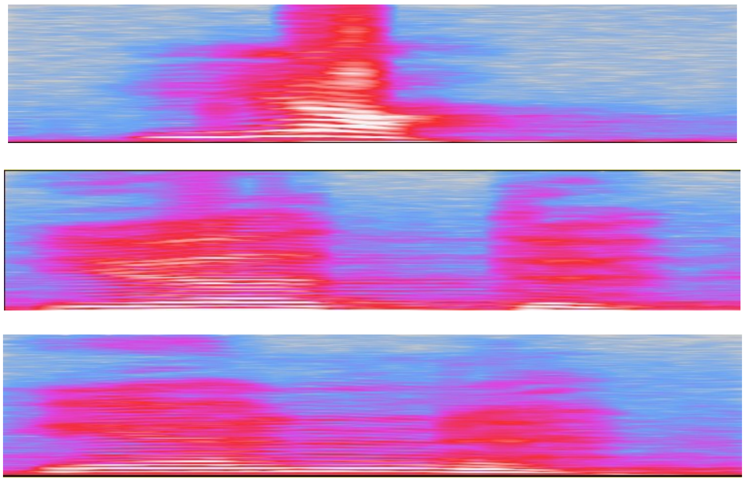


Figure 2. Spectrogram alphabet ya, yab, and yad

3.1.3. MFCC

Introduced by Davis and Mermelstrein in 1980 [2], MFCCs are calculate a follow [14]:

- Frame the signal into short frames;
- Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows;
- Take the logs of the powers at each of the mel frequencies;
- Take the discrete cosine transform of the list of mel log powers, as if it were a signal;
- The MFCCs are the amplitudes of the resulting spectrum.

The image produced by these Pre-processing steps is then fed into multi-layer convolutional neural networks, with a fully-connected layer followed by a softmax at the end

4. CONVOLUTIONAL NEURAL NETWORKS

4.1. The perceptron

Perceptron is a very simple learning machine algorithm based on a model of biological neurons, which takes an input vector, weigh matrice, and an activation function to produce the desired output [15, 16]. The weights are the property of the connection which represent the strength of the connection. Each connection has a different weight value while bias is the property of the neuron as shown in Figure 3.

4.2. The multilayer perceptrons MLP

When we combine many perceptrons, we form a multilayer perceptron or more precisely an artificial neural network [15]. The first layer is the input layer, corresponding to the data features. The last layer is the output layer, which provides the output probabilities of classes or labels as shown in Figure 4.

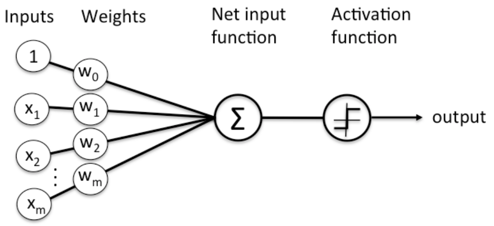


Figure 3. A perceptron

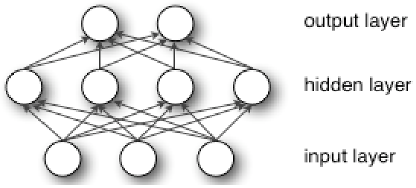


Figure 4. The multilayer perceptrons

4.3. CNN

The CNNs or ConvNets these are a particular form of neural network [17, 18] that takes an input image inspired by the work of Hubel and Wiesel on the primary visual cortex of the cat [19] as shown in Figure 5. The CNN architecture has two components: the convolutive part or feature extraction part, we use spectrogram technique to extract the feature. And the classification part, the vector of feature extracted by the convolutive part is feed to the fully connected layers leading into the output layer which represents the classifier. The convolutive part consists of [5-20].

Convolutional layer: convolution is one of the main building blocks of a CNN. based on its convolutional mathematical principle [21], is consists of a set of learnable filters, or kernels. Each filter is applied by independently striding over the entire input, creating an output feature map for each filter. At every location, a matrix multiplication is performed and sums the result onto the feature map as shown in Figure 6.

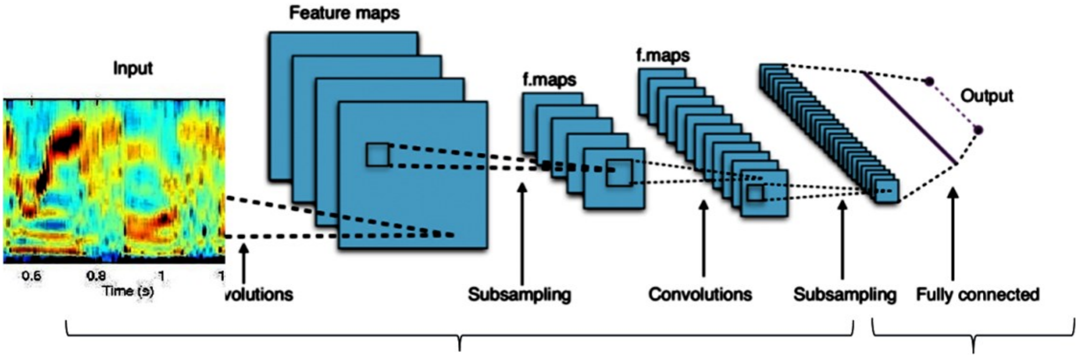
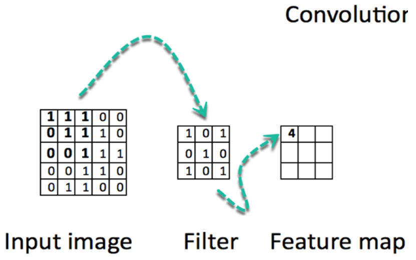


Figure 5. Illustration of the architecture used for the CNN with many layers



1	**1**	**1**	0	0
0	**1**	**1**	1	0
0	**0**	**1**	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

4		

Figure 6. Convolutional mathematical principle [20]

Pooling layer: reduces the size of the image. It is an essential layer often placed between two layers of convolution. The ReLU correction layer therefore replaces all negative values received. The classification part consists of fully connected layers. Fully-connected layers are used after the final convolutional layer in order to match the output size of the neural network to the desired output size. The output is then passed through a softmax function in order to create a probability representation for the predictions for each class in the supervised learning setting. To use the fully-connected layers, the output from the final convolutional layer is commonly flattened out, or the feature maps are subsampled to a size of 1.

CNN Architecture:

In this paper the CNN uses tow convolutional layers:

- Convolutional layer with 64-8 and 20 filters
- Relu layer
- Max pooling layer %with 2×2 filter
- Convolutional layer with 64-4 and 10 filters
- Max pooling layer %with 2×2 filter
- Relu layer

5. TENSORFLOW

TensorFlow is an open source library developed by Google's AI organization, as a middlewear library that can be used to build deep learning neural networks, TensorFlow is written in python and c++ with a model and robust archetecture that can be run on multiple CPUs and GPUs [22] as shown in Figure 7. Speakers and the test has with Tensor Flow, machine learning algorithms are based on the concept of the data flow graph or computional graph [23]. The nodes of this graph represent mathematical operations. The edges are tensors. In terms of TensorFlow, a tensor is just a multi-dimensional array. Each data flow graph computation runs within a session on one or more CPUs or one or more GPUs.

A computational graph in TensorFlow consists of several parts:

- Tensor: a multi-dimensional array.
- Graph: a central hub that connects all the variables, placeholders, constants to operations.
- Constants: are fixed value tensors-not trainable.
- Variables are tensors initialized in a session-trainable.
- Placeholders: are tensors of values that are unknown during the graph construction, but passed as input during a session.
- Operations: are functions on tensors.
- Session: A session creates a runtime in which operations are executed and Tensors are evaluated.

We opted for TensorFlow for the following reasons: TensorFlow comes with a complete set of visualization tools that make it easy to understand, debug, and optimize applications. TensorFlow also has a large community of users and lots of documentation.

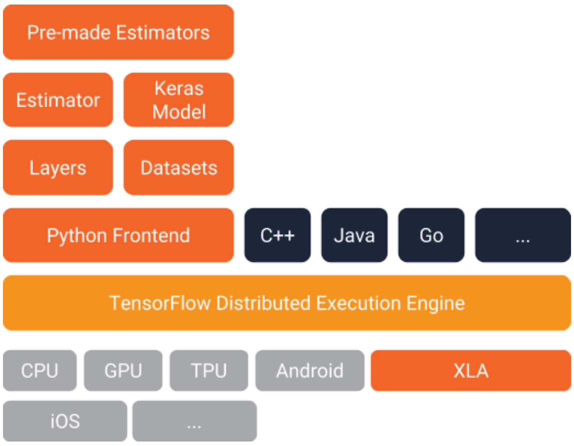


Figure 7. TensorFlow architecture released by Google for implementing the machine learning models

6. EXPERIMENTS AND RESULTS

6.1. Environment

We choose to install TensorFlow with GPU, use virtualenv installation on a workstation hp Z640 Intel 4 core, and we use Linux Ubuntu 16.04, to avoid the problem [23]. The following software needs to be installed properly [24]:

- pip and Virtualenv;
- CUDA Toolkit 9.0;
- GPU card with Compute Capability (CUDA) 3.0;
- GPU drivers;

- cuDNN SDK v7.
- Priore to installing TensorFlow with GPU support, ensure that the system support all NVIDIA software requirements.

6.2. Corpus

To train our model we use the dataset collected by Satori H, and al [10]. The signals were recorded in a non-noisy space by the same microphone; the recording files are in MS WAV format with a specific sample rate–16 kHz, 16 bit mono. Each speaker was invited to pronönce 33 Amazigh letters 10 times. During training, the corpus is separated into:

- Training data: 80% of the data;
- Validation data:10% of the data is reserved for the evaluation of the precision during the training;
- Data tests: 10% of the data is used to evaluate accuracy once the training is complete.

In the following Table 1. We define how many audio files used in training, validation, and test data for 3 separated experiments described in this paper.

Table 1. Training, validation, and test data for the 3 Experiments

	Corpus (audio file)	Training	Validation	Test
Experiment 1	9240	7392	924	924
Experiment2: Females	9240	7392	924	924
Experiment 2: Mals	9240	7392	924	924
Experiment 3: age 9-15	4620	3696	462	462
Experiment 3: age 16-30	4620	3696	462	462
Experiment 3: age +30	4620	3696	462	462

6.3. Train CNN with TensorFlow

Basically, there are 3 steps to build a CNN model in Tensorflow:

- Preprocessing the data;
- Build the model; build the nodes and operations and how they are connected to each other;
- Train and estimate the model on some data;

We feed our CNN by spectorgram results from the preprocessing phase to train and predict the labels. The labels used in this paper are «silent", "ya", "yab", "yad"... Each column represents a set of samples that was predicted to be each, so the first column represents all the clips that were intended to be silence. The second column represents all those that were predicted to be ya word, and the third "yab" [14]. At the end of the training. A final confusion matrix will generate. The columns of this matrix represent the prediction labels and the lines represent the actual labels. So, she gives a good summary of training errors.

6.3.1. Experiment 1

The corpus consists of 9240 audio files, 28 speakers was invited to pronounce 33 letters Amazigh 10 times. The corpus is divided into 7392 train, 924 test, and 924 validation audio files. The results show that the system produces 89.8% of accuracy as shown in Figure 8.

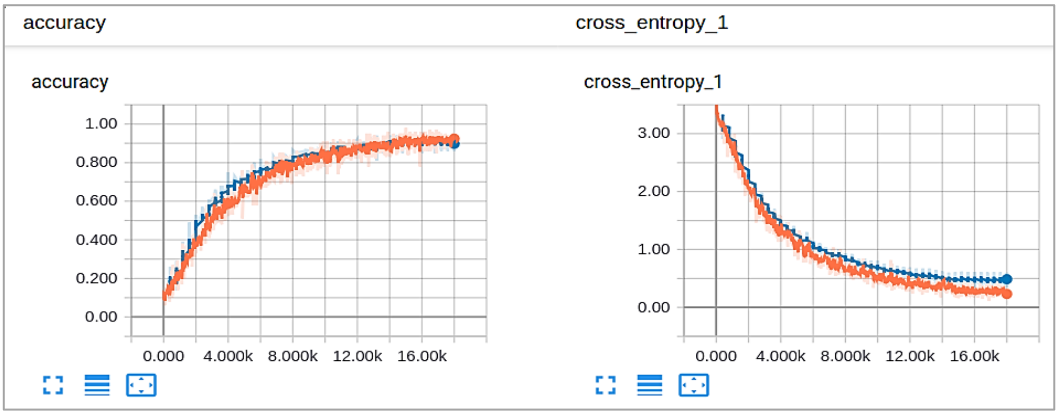


Figure 8. A graph showing the training CNN models progress

6.3.2. Experiment 2

In order to test the effect of the gender on the quality of the acoustic model, the corpus consists of 9240 audio files: 14 females and 14 females' speakers were invited to pronounce 33 letters Amazigh 10 times. In the following Table 2 we defined how many audios files training, validation, and test data, and result. The results show that the best results were recorded for males with 93.8% of accuracy.

Table 2. Recognition accuracy for experiment2: corpus consists of 9240 audio files

	Corpus	Training	Validation	Test	Accuracy%
Females 'Corpus	4620	3696	462	462	93.3
Males 'Corpus	4620	3696	462	462	93.8

6.3.3. Experiment 3

In this experiment, we evaluate the performance of a system, which was trained and tested for different age. The corpus consists of 13860 audio files: 42 speakers was invited to pronounce 33 letters Amazigh 10 times, we have classified the speakers 'ages into three categories: age 9-15, age 16-30, and age +30. In the following Table 3 we defined how many audios files training, validation, and test data, and result. The results show that the best results were recorded for +30 age category. To test the effect of sex or age variation on the quality of the system, it has been trained and tested with different corpuses our results are already encouraged, the best results produce 93.9% of accuracy.

Table 3. Recognition accuracy for experiment3: corpus consists of 13860 audio files

Age	Corpus	Training	Validation	test	Accuracy%
9-15	4620	3696	462	462	91.4
6-30	4620	3696	462	462	92.4
+30	4620	3696	462	462	93.9

6.3.4. Comparative analysis

The presented work has been compared with the existing similar task recognition, especially, the emotion recognition system SER and sound event recognition. The following Table 4 lists a number of results from our previous work [3], Zheng W. Q., *et al.* [25], Zhang H., *et al.* [26], and our Proposed. In our previous work [3] we have developed the Amazigh speech recognition system based on hidden Markov model HMMs using the open source CMU Sphinx-4. The corpus consists of 11220 audio files. The system obtained best performance of 90 % when trained using 128 Gaussian mixture models, and 5 number of HMMs states. Zheng W. Q., *et al.* [24], have developed the emotion recognition system based on convolution neural networks with 2 convolutions+2 pooling layers, and using labelled training audio data and used the log-spectrogram to extract feature, component analysis PCA to reduce the dimensionality. The system achieved about 40% accuracy. Zhang H., *et al.* [25], have proposed the sound event detection based on convolution neural networks with 2 convolution +2 pooling layers, and spectrogram to extract feature. The system the system achieved about 94.07% of accuracy. In our proposed work; the system obtained the best performance of 93.9% of accuracy when trained using +30 age category. Results are very satisfactory if compared with the existing similar works.

Table 4. Tabular comparison of the recognition accuracy of the proposed systems

	Approch	Corpus	Accuracy%
	MFCC-HMM	11220	90
	Spectr+CNN (2 Conv+2 pool)	5288	33
	Spectr+PCA+CNN (2 Conv+2 pool)	5288	40
	MFCC +CNN (2 Conv+2 pool)	4000	94.07
Exp:1	MFCC +CNN (2 Conv+2 pool)	9240	89.8
Exp:2Females	MFCC +CNN (2 Conv+2 pool)	4620	93.3
Exp:2Mals	MFCC +CNN (2 Conv+2 pool)	4620	93.8
Exp:3Age9-15	MFCC +CNN (2 Conv+2 pool)	4620	91.4
Exp:3Age16-30	MFCC +CNN (2 Conv+2 pool)	4620	92.4
Exp:3Age+30	MFCC +CNN (2 Conv+2 pool)	4620	93.9

7. CONCLUSION

In this work, a simple machine learning model that recognizes a 33 Amazigh letters using CNN and GPU computation was developed. We followed MFCC Technique to extract the features. The system was built with open source TensorFlow. The system obtained the best results when trained using +30 age category produced 93.9% of accuracy. Our work and among the first contributions address the Amazigh ASR based on convolutional neural networks and GPU computation. In perspective, to overcome the overfitting problem and improve the performance, we use a Dropweak Regularization for CNNs in Amazigh recognition system dropweak is based on the idea of dropping the weak weights in a neural network, we mean setting its value to zero so he cannot influence on the output of its processing unit.

REFERENCES

[1] M. Hamidi, *et al*., “Amazigh digits through interactive speech recognition system in noisy environment,” *International Journal of Speech Technology*, vol. 23, no. 1, pp. 101-109, December 2020.

[2] M. Telmem, *et al.,* “Amazigh speech recognition system based on CMUSphinx,” *Proceedings of the Mediterranean Symposium on Smart City Applications Springer*, pp. 397-410, January 2017.

[3] O. Zealouk, *et al*., “Vocal parameters analysis of smoker using amazigh language,” *International Journal of Speech Technology*, vol. 21, no. 1, pp. 85-91, 2018.

[4] M. Telmem, M., *et al*., “Estimation of the optimal HMM parameters for amazigh speech recognition system using CMU-sphinx,” *Procedia Computer Science*, vol. 127, pp. 92-101, 2018.

[5] A. Bhandare, *et al.,* “Applications of convolutional neural networks,” *International Journal of Computer Science and Information Technologies*, vol. 7, no, 5, pp. 2206-2215, 2016.

[6] D.Palaz, *et al.,* "End-To-End Acoustic Modeling Using Convolutional Neural Networks For HMM-Based Automatic Speech Recognition," *Speech Communication*, vol. 108, p. 15-32, 2019.

[7] S. K. Mohapatra, *et al*., “Diabetes detection using deep neural network,” *International Conference on Soft Computing Systems*, vol. 4, no. 4, December 2018, pp. 243-246.

[8] H. K. Palo, *et al*., “Emotion analysis from speech of different age groups,” *Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering*, vol. 10, June 2017, pp. 283-287.

[9] H. Zhang, *et al*., “Comparison on neural network based acoustic model in mongolian speech recognition,” *Asian Language Processing (IALP)*, November 2016.

[10] H. Satori, *et al*., “Voix comparaison entre fumeurs et non-fumeurs a l'aide de la reconnaissance vocale HMM Système,” *International Journal of Speech Technology*, vol. 20, no. 4, pp. 771-777, 2017.

[11] M.Ameur, *et al*., “Initiation la langue amazigh,” *Institut Royal de la Culture Amazighe*, 2004.

[12] X. Liu, “Deep convolutional and LSTM neural networks for acoustic modelling in automatic speech recognition,” pp. 1-9, 2018.

[13] K. Kumar, *et al*., “A Hindi speech recognition system for connected words using HTK,” *International Journal of Computational Systems Engineering*, vol. 1, no 1, pp. 25-32, 2012.

[14] B.J. Mohan, et al., "Speech Recognition Using MFCC and DTW," 2014 *International Conference on Advances in Electrical Engineering (ICAEE),* IEEE, 2014, pp. 1-4.

[15] S. Vieira, S, *et al*., “Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: methods and applications,” *Neuroscience Biobehavioral Reviews*, no. 74, pp. 58-75, 2017.

[16] M. Telmem, *et al*., “A comparative study of HMMs and CNN acoustic model in amazigh recognition system,” *Embedded Systems and Artificial Intelligence. Springer*, pp. 533-540, April 2020.

[17] O. Abdel-Hamid, *et al*., “Convolutional neural networks for speech recognition,” *ACM Transactions on audio, speech, and language processing*, vol. 22, no 10, pp. 1533-1545, October 2014.

[18] K. Krishna, *et al*., “A study of all-convolutional encoders for connectionist temporal classification,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[19] D. H. Hubel, *et al*., “Receptive fields, binocular interaction and functional architecture in the cat's visual cortex,” *The Journal of physiology*, vol. 160, no 1, pp. 106, January 1962.

[20] A. Khan, *et al*., “A survey of the recent architectures of deep convolutional neural networks,” *Artificial Intelligence Review*, 2020.

[21] Brandon Rohrer, “How convolutional neural networks work,” [Online]. Available: https://brohrer.github.io/howconvolutionalneuralnetworkswork: html.

[22] [Online]. Available: https://www.tensorow.org/tutorials/sequences/audiorecognition.

[23] Goldsborough, P., “A tour of Tensorow,” *arXiv preprint arXiv:1610.01178*, 2016.

[24] TensorFlow, “Install TensorFlow 2,” [Online]. Available: https://www.tensorow.org/install/.

[25] W. Q. Zheng, *et al.,* “An experimental study of speech emotion recognition based on deep convolutional neural networks,” *2015 International conference on a_ective computing and intelligent interaction (ACII),* September 2015, pp. 827-831.

[26] H. Zhang, *et al*., “Robust sound event recognition using convolutional neural networks,” *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, pp. 559-563.